Day 5: Looping, visualization & Assessment

What you have learned so far in week 1 …

- Connecting to a compute cluster (AWS or BioFrontiers Fiji)
- Working on a Unix command terminal
- Running scripts processing FASTQ sequencing files
  - Quality control (FASTQC)
  - Trimming (Trimmomatic)
  - Mapping (HISAT2) to yield SAM/BAM files

What you have learned so far in week 1 …

- Connecting to a compute cluster (AWS or BioFrontiers Fiji)
- Working on a Unix command terminal
- Running scripts processing FASTQ sequencing files
  - Quality control (FASTQC)
  - Trimming (Trimmomatic)
  - Mapping (HISAT2) to yield SAM/BAM files

- Goals for today:
  - Part 1:
    - `for` loops
    - Transforming mapped files to smaller visualization files
  - Part 2: Assessment of skills learned up through today

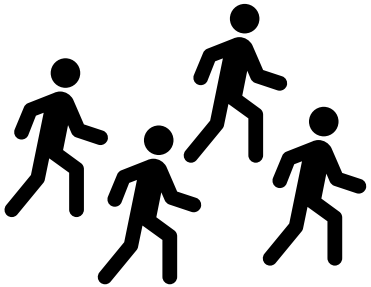
`for` loop demonstration

FOR_loop_basics.md worksheet under Day 5 worksheets

- This worksheet is for your reference if you need it, though you can follow along right now if you want to

- Goes over serial and parallel `for` loops with SLURM

Contestants to go through code (obstacle course). There are multiple.



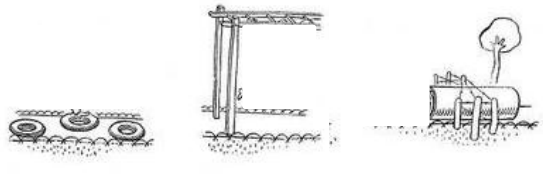
Fastq

Fastq

Fastq

Fastq

The code (obstacle course) that stays the same and we want to reuse.

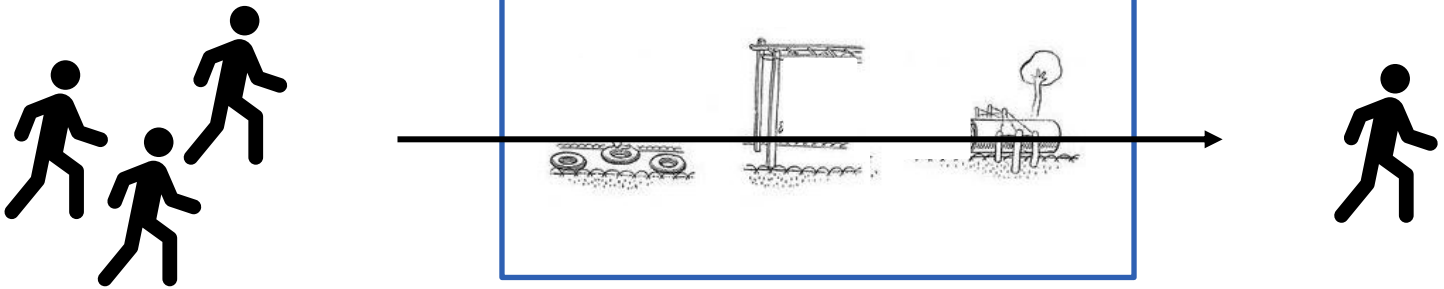


CPU

Script: processing.sbatch

Contestants to go through code (obstacle course). There are multiple.

The code (obstacle course) that stays the same and we want to reuse.



CPU

Fastq

Fastq

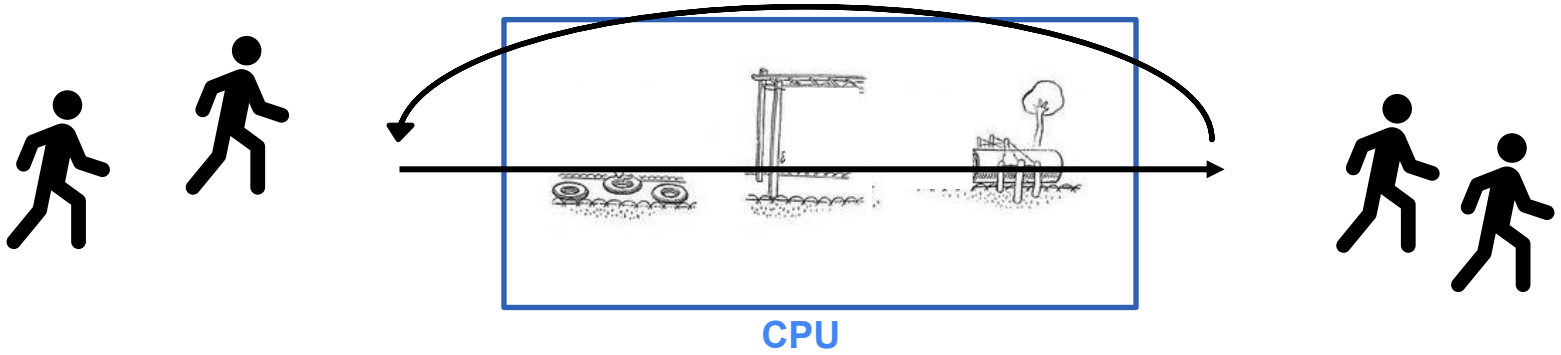
Fastq

Script: processing.sbatch

TDF

Contestants to go through code (obstacle course). There are multiple.

The code (obstacle course) that stays the same and we want to reuse.



CPU

Fastq

Fastq

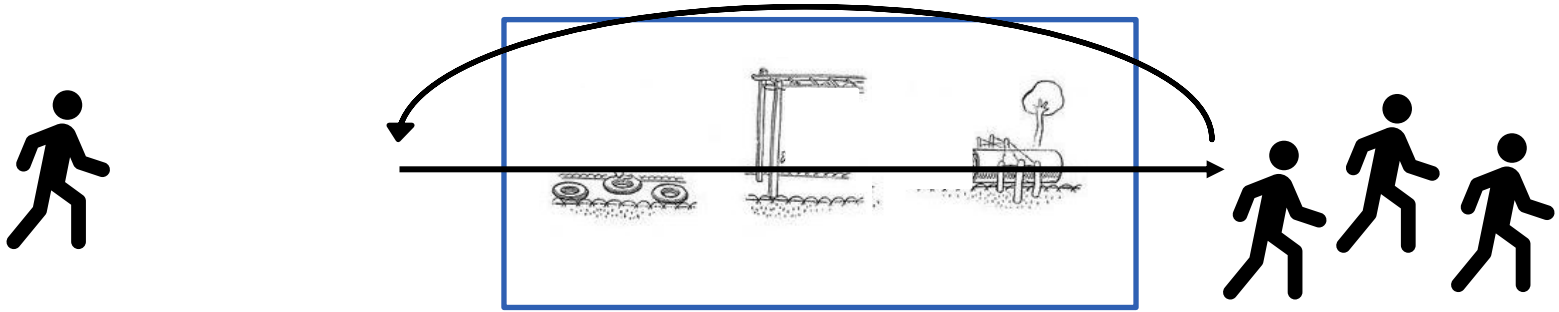
Script: processing.sbatch

TDF

TDF

Contestants to go through code (obstacle course). There are multiple.

The code (obstacle course) that stays the same and we want to reuse.



CPU

Fastq

Script: processing.sbatch

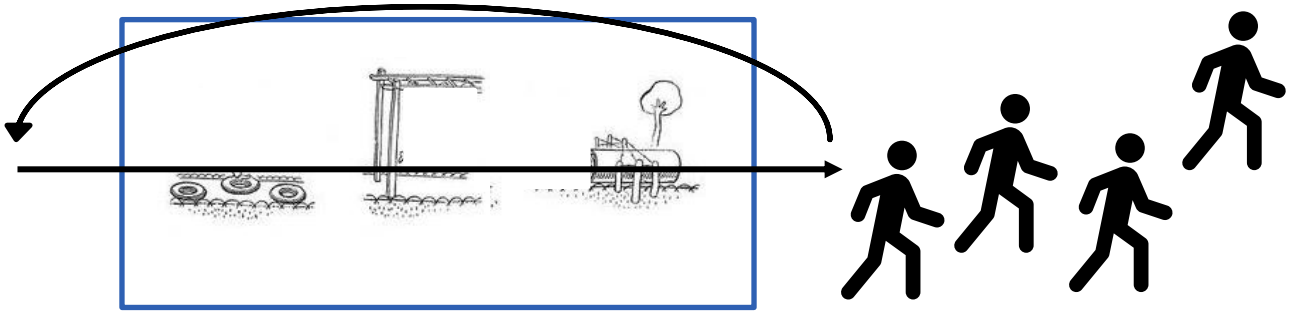
TDF

TDF

TDF

Contestants to go through code (obstacle course). There are multiple.

The code (obstacle course) that stays the same and we want to reuse.



CPU

Script:
processing.sbatch

TDF

TDF

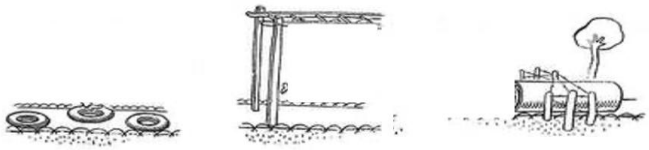
TDF

TDF

Script:
run_processing.sh

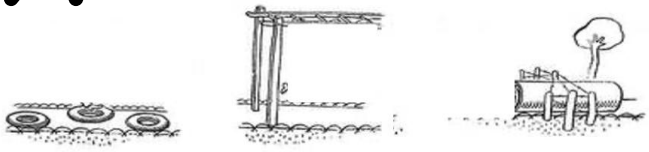
Fastq

Script:
processing.sbatch



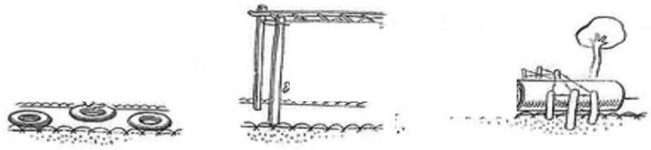
Fastq

Script:
processing.sbatch



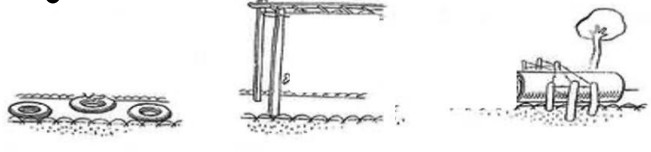
Fastq

Script:
processing.sbatch



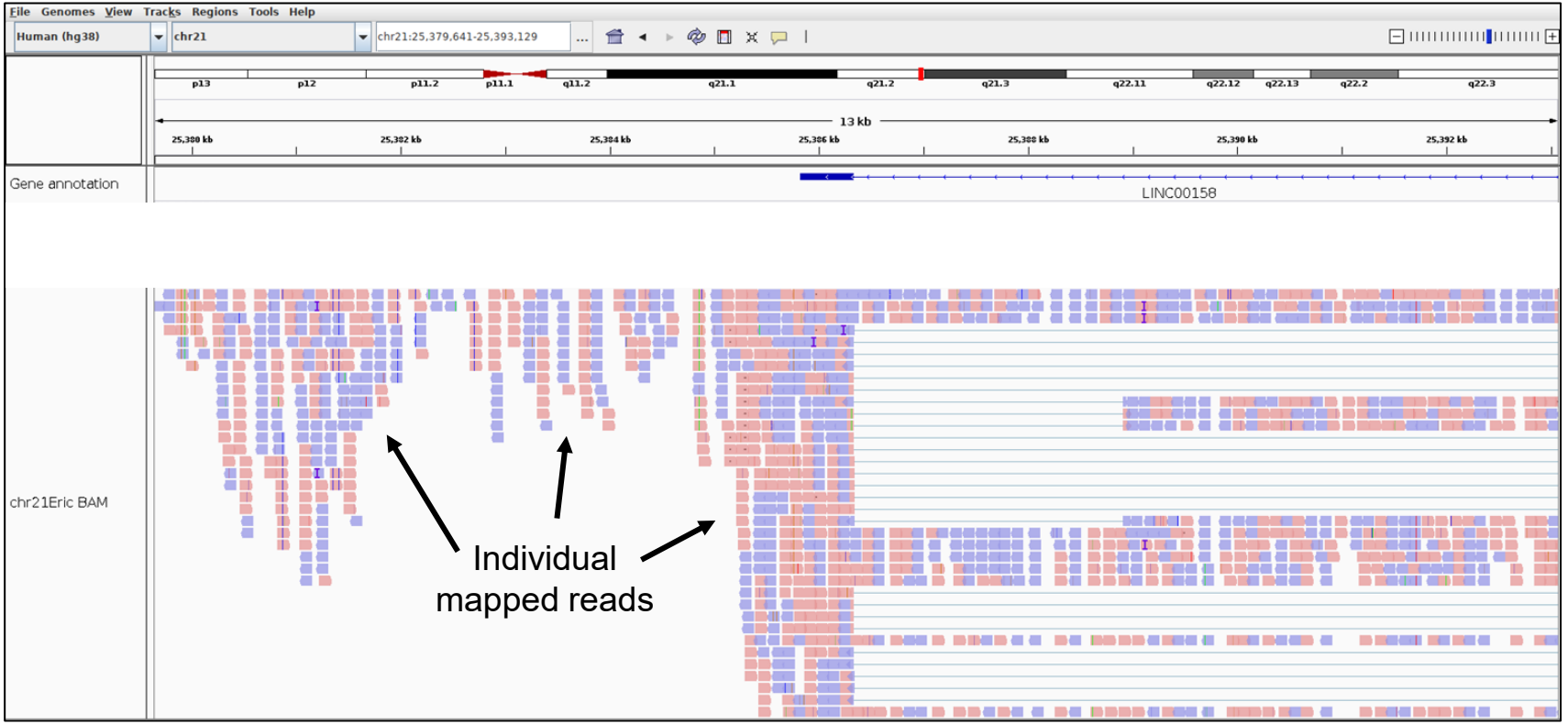
Fastq

Script:
processing.sbatch



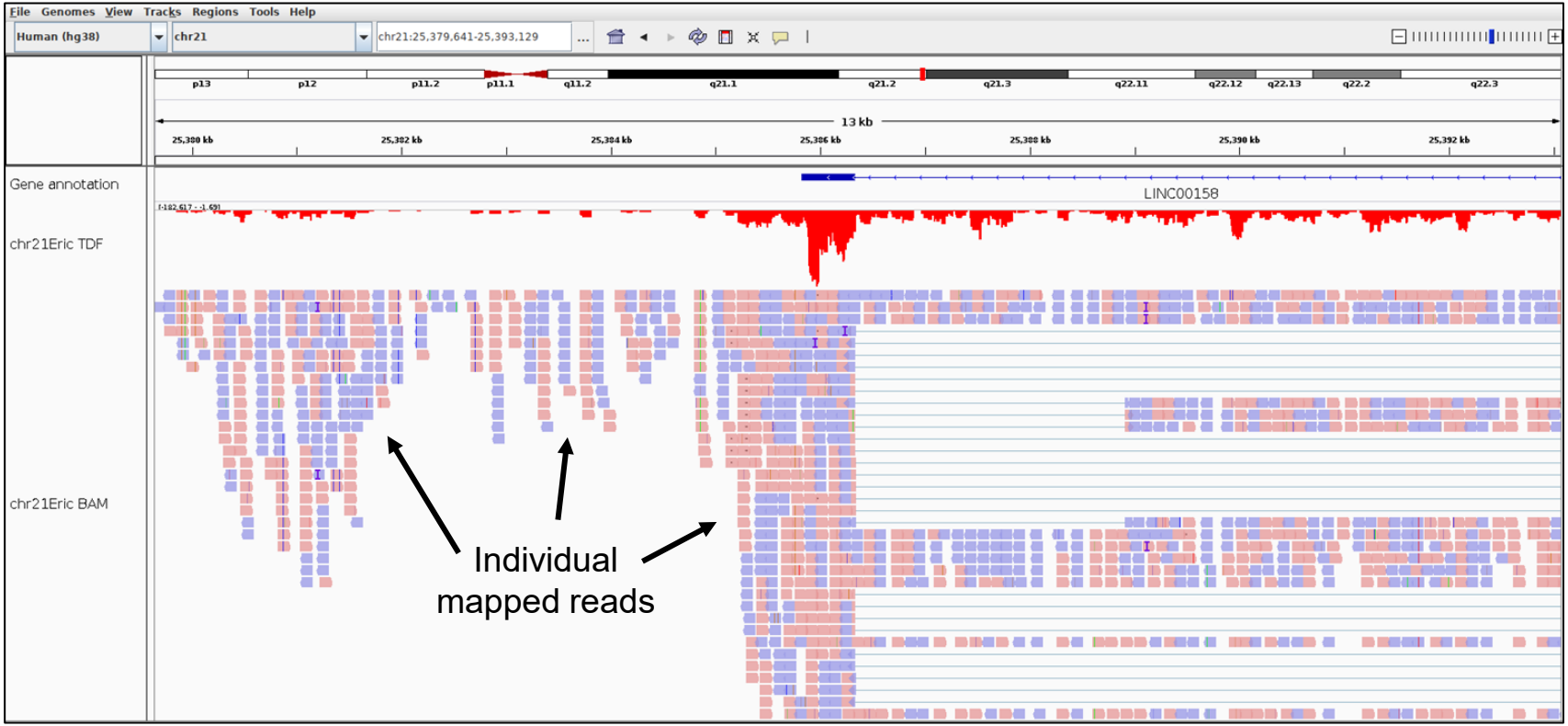
Comparing BAM and TDF files on IGV

Observed region on screen = **13 kb**



Comparing BAM and TDF files on IGV

Observed region on screen = **13 kb**



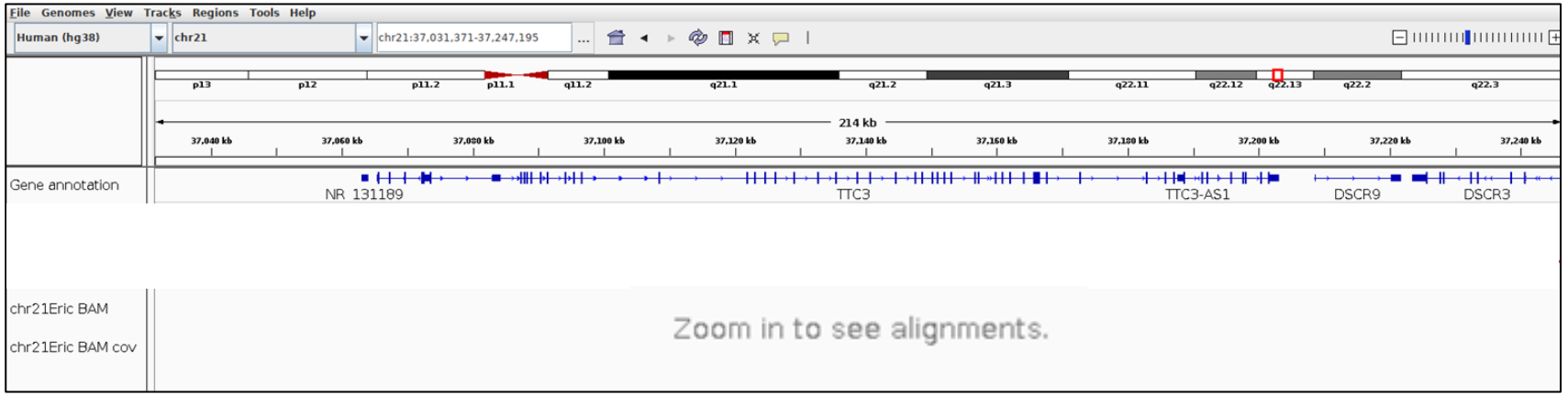
Comparing BAM and TDF files on IGV

Observed region on screen = **214 kb**

annotation

TDF

BAM



IGV does not display individually mapped reads on such a big region at once!

But IGV does okay displaying TDF coverage across any zoom region.

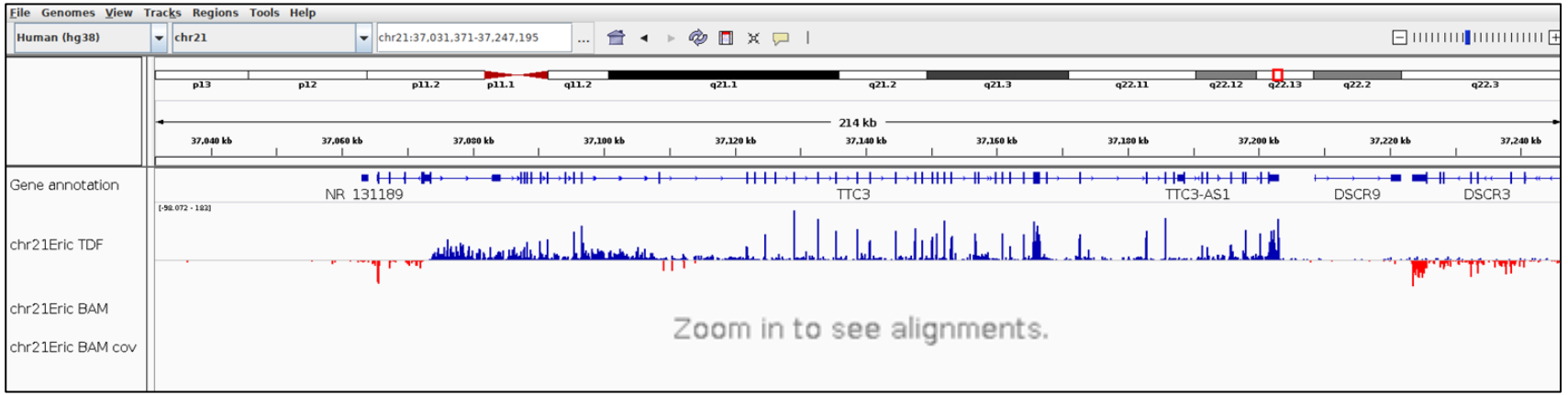
Comparing BAM and TDF files on IGV

Observed region on screen = **214 kb**

annotation

TDF

BAM



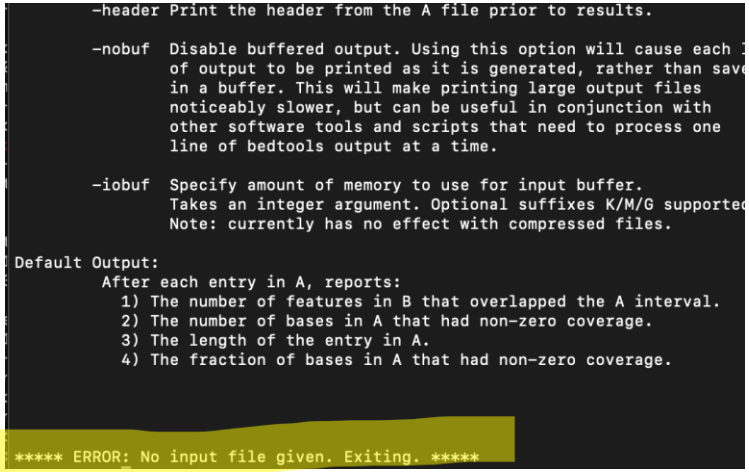
IGV does not display individually mapped reads on such a big region at once!

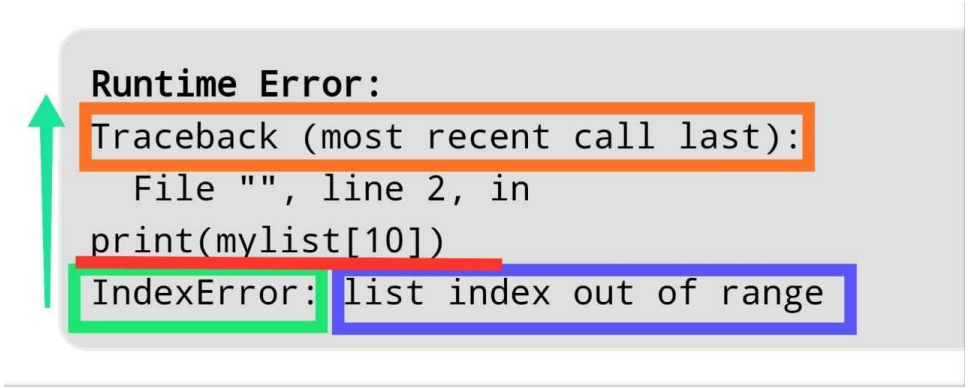
But IGV does okay displaying TDF coverage across any zoom region.

Go to the Day5 worksheet on Github, and get started on today’s exercise

First things to check when error

1. Did you check what the error and output files said?
    1. If you can't find them, check that you fixed the path in the script for these files
2. Reading an error:
    1. Usually the most helpful part of the error is at the bottom of it
3. TYPOS (especially filenames and folders)





Day 5: Assessment

Check out the other FASTQ datasets in Day5.

Imagine these files are fresh off the sequencer… can you write your own scripts to turn them from raw FASTQs to TDFs? What QC checks should you perform?

(Extra-very-real-points: Do all files with a loop!)

FASTQ → SAM → BAM →
→ BEDGRAPH → TDF

First things to check when error

1. Did you check what the error and output files said?
    1. If you can't find them, check that you fixed the path in the script for these files
2. Reading an error:
    1. Usually the most helpful part of the error is at the bottom of it
3. TYPOS (especially filenames and folders)